Increasing XAI trustworthiness: Optimising Explanation Quality through Additional Loss Functions.

GERALD FREISLICH[1], **BERNARD EVANS**[1]
[1]School of Computer and Mathematical Sciences, The University of Adelaide, Australia

E-MAIL: {gerald.freislich, bernard.evans}@adelaide.edu.au

Abstract:

Modern Neural Networks are capable of preforming a wide variety of complex tasks, often with commendable accuracy. These Neural Networks are black-boxes, where the methods through which the network forms its predictions are abstracted away from the user. To provide insight into how and why a Neural Network formed its prediction, eXplainable Artificial Intelligence (XAI) was developed. When used on a Neural Network, XAI depicts what, in the input, informed the Neural Network's prediction. These explanations of the Neural Network's behaviour, can however, be low quality which limits the usefulness and trustworthiness of the XAI. Here, we introduce Additional Loss Functions as a novel technique to ensure that XAI, such as Integrated Gradients (IG) and Class Activation Maps (CAM), provide high quality explanations. We facilitate improved explanation quality by embedding explanation quality evaluation into the training of the Neural Network. By doing this, we find that the quality of explanations from IG and CAM can be optimised for three common quality metrics, at no cost to the underlying Neural Network's accuracy. With two of the optimised models we also attain a higher macro-F1 than the latest benchmark Graph Neural Networks achieve on CORA.

Keywords:

Machine Learning, Neural Networks, XAI, Graph-based Machine Learning

1 Introduction and Motivation

Consider a social network of individuals who are and are not friends. We can represent this as a graph, where individuals are nodes and they have edges to their friends - if any. Some individuals of this graph are part of the Computer Science (CS) Club, and some are not. To market an upcoming event, we want to know who is a part of the CS Club, and for this we train a Graph Neural Network (GNN) on a subset of the graph. After training, the GNN can predict who is in the CS Club with an accuracy of 93%. When reviewing the GNN's performance, we find that a node misclassified as not part of the CS Club is the club-president. In order to understand why the GNN misclassified the club-president, XAI is utilised.

In this scenario we observe the usefulness of XAI. But what is XAI and why is it necessary? Firstly, some Machine Learning (ML) models, like Neural Networks (NN), are black-boxes, as the decisions they make are hidden [13]. Problematically, NN's are very useful, and their application to complex tasks has recently increased exponentially [6]. The need to understand how these black-box models function is resolved by eXplainable Artificial Intelligence (XAI) [17]. Explainability and XAI is by definition "a set of processes and methods that allows humans...to comprehend and trust the results and output created by machine learning models" [7]. This trust and transparency induced on ML models by XAI has become a "non-functional requirement" [15], with the EU stipulating it's necessity for the ongoing use of Machine Learning [11]. As such, good XAI is imperative to the continue use of Machine Learning techniques.

But how can we ensure that XAI is good? What if the XAI provides an explanation of the GNN's behaviour that simply does not make sense? What if the XAI, in the scenario, declares that the Club President's age of 21, while all other individuals are 18-22, was the most significant characteristic that the GNN used to make its prediction? To satisfy this problem, explainability metrics have been created. These XAI metrics provide insight into the trustworthiness of XAI by evaluating their quality, with respect to some criteria [3]. A natural question then appears: Can we ensure high quality explanations from a given XAI according to a set of metrics. That is; *Can we train a Neural Network to have high quality explanations from a given XAI*. This question underpins the following investigation.

In this paper we consider three research questions, which direct the investigation.

RQ1: Can we ensure higher quality explanations from XAI through training to optimise explanation quality.

RQ2: What are the XAI metrics that can be best optimised through Additional Loss Functions.

RQ3: Are there reductions in accuracy of the Neural Network when training to optimise XAI quality alongside accuracy.

We find that a higher quality of explanations from Integrated Gradients and Class Activation Maps can be attained at no expense to the accuracy of the Graph Neural Network on two common node classification tasks: CORA and Citeseer.

The paper is structured as follows: first the relevant knowledge of the XAI used, and XAI quality metrics are provided. Then, the framework is addressed, followed by the results of the experiments, and then a discussion of the modus operandi by which the framework functions. We finish with the conclusions and future work.

2 Relevant Literature

In Section 1, we introduced the problem of poor quality explanations in XAI. To improve XAI quality on a Neural Network, we instill XAI quality evaluation in the training of the network. Here, we define the XAI and XAI quality metrics used in this investigation.

2.1 XAI models

The XAI used in this investigation is Integrated Gradients (IG) and Class Activation Maps (CAM). Here, we review the theory behind each XAI. To perform node classification, Graph Neural Networks (GNN) learn the structure of the graph [21]. Resultantly, when making a prediction on the class of a given node *u*, a GNN will be influenced by the connections of *u*. This occurs due to an assumption of homophily, where like nodes communicate and are connected to other like nodes. In the scenario, in Section 1, a CS Club member is likely to communicate with other CS Club members, and these connections are expected to be important to the classification of the original member as part of the CS Club. As a node's connections influences its classification by a GNN, XAI like IG and CAM provide a value quantifying the importance of each connection that a given node *u* has. This value of importance for a node *v* in another node *u*'s classification is the *attribution* of *v* on *u*.

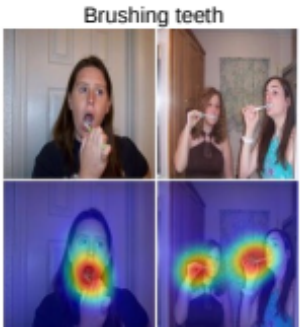

FIGURE 1. Example of CAM heat-map for classifying the action of 'Brushing Teeth' [20]

2.1.1 Integrated Gradients

Proposed in 2017 by Sundararajan et al. [16] Integrated Gradients measures the importance of features in the prediction of Neural Networks. First used in Computer Vision the technique takes the gradient between a Baseline vector x' and the given Input vector x for all points in the vector. Formally, the definition of Integrated Gradients is:

$$IG_i(x) ::= (x_i - x_i') \times \int_{a=0}^{1} \frac{\partial F(x' + a \times (x - x'))}{\partial x_i} \quad (1)$$

IG resolves two fundamental problems of earlier techniques: Sensitivity and Implementation Invariance. Sensitivity is the notion that "if every input vector and baseline vector differ in one value, and the prediction is resultantly changed" then the feature that is different between the two vectors is important. Implementation Invariance, however, is the belief that two Neural Networks with the same outputs for all inputs are functionally the same, and hence should receive the same explanations. Integrated Gradients was introduced into Graph-based Machine Learning in 2019 by Wu et al. [18] and has since been used extensively. Nonetheless, there are problems with IG, namely the effect of saturation where the model's explanations can become more sensitive to noise [9].

2.1.2 Class Activation Maps

Class Activation Maps "indicates discriminative image regions" used by a Neural Network "to identify...category"[20]. Originally used in images to depict what pixels had the greatest importance in a Neural Network's prediction, CAM has been developed extensively over time, with popular variations like GradCAM [14] and Eigen-CAM[10]. We observe the heat-map of CAM in Figure 1 which depicts the importance of each pixel

in the classification of the image as “Brushing Teeth”. CAM operates by summing the weights associated with a class multiplied by the output of the Neural Network for that given class, “directly indicat[ing] the importance of the activation...leading to the classification”[20] of the image. This is formalised in Equation 2.

$$M_c(x, y) = \sum_k w_k^c f_k(x, y) \tag{2}$$

2.2 XAI metrics

Here, we review the three XAI metrics used in the investigation; which are sufficiency, stability and sparsity. For the calculation of each metric, and XAI quality, we assume that explanation vectors from each XAI belong to the regular Euclidean space, with Euclidean distance.

2.2.1 Sufficiency

For XAI there is a notion of explanation Faithfulness, that is: ‘are the explanations from an Explainability model faithful to the underlying Neural Network’ [4]. This ‘faithfulness’ evaluates whether the top q% of values in an explanation are representative of what the Neural Network is learning. For instance, if XAI presents that only three features in an input were important for the Neural Network’s decision, and we set all the other features to zero - and cause a prediction change - then those top three features were not representative of what the model learnt. Hence, the XAI is unfaithful to the Neural Network. An example of a Faithfulness metric is sufficiency, as proposed by DeYoung et al. [5]. This metric is provided in equation 3.

$$SUFF = \frac{1}{|B|} \sum_{q \in B} (p_{c(x)}(x) - p_{c(x)}(x_{:q\%})) \tag{3}$$

Here, B is the set $B = \{1, 5, 10, 20, 50\}$. Each of these values is a percentage of the top q% tokens to leave unmolested [4], and all other tokens in the explanation are set to zero. By setting all other tokens to zero we get $x_{:q\%}$, and the original input is x. Further, p is the Neural Network, and for sufficiency we take the Euclidean distance between the two prediction vectors x and $x_{:q\%}$.

2.2.2 Stability

Stability is a metric that uses small perturbations on the input data for its [19]. These perturbations, as shown by Agarwal et al. 2022 [2], are formed through the random addition or subtraction of edges from the input graph. Formally stability is:

$$Stability = D(E(u), E(u')) <= \delta, \tag{4}$$

where $E(u')$ is the explanation for an perturbed node u, and $E(\cdot)$ is the XAI. $D(\cdot)$ is an distance function, and δ is a constant. In our case, we neglect δ as we wish to optimise its value, and hence a threshold is arbitrary. For the distance function, we create matrices from the explanation vectors of the perturbed and unperturbed graph (M_u and $M_{u'}$ respectively) and find the difference between the two frobenius norms of the two matrices. Our Stability function is defined in Equation 5.

$$Stability = ||M_u||_F - ||M_{u'}||_F \tag{5}$$

2.2.3 Sparsity

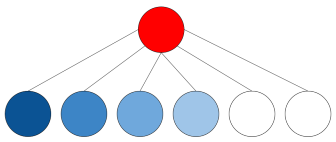

FIGURE 2. Example of a sparse explanation in Graph-based Machine Learning, where a given node’s classification (red) is dependent on some of the nodes in its subgraph, but not all under the presumption of locality.

Sparsity is a metric for locality, under the assumption that XAI which produces fewer non-zero feature importances is better than one with many [19]. A better XAI, according to Sparsity, will select a handful of important features and ’ignore the rest’ [19]. Provided by Pope et al. [12], Sparsity is the proportion of non-zero importances in an explanation vector against the vector’s size (refer to equation 6).

$$Sparsity = \frac{1}{|V|} \sum_{u \in V} (1 - \frac{|m_u|}{|M_u|}), \tag{6}$$

where m_u is the set of non-zero importances for a given node u, and M_u is the set of all importances for a given node u.

We can see, in Figure 2, a node classification that is sparse, where not all members of the nodes’ subgraph have non-zero attributions. This indicates, that the remaining two nodes on the right, where not important to the node’s classification, whilst the node on the left was the most important to the red nodes classification.

3 Framework

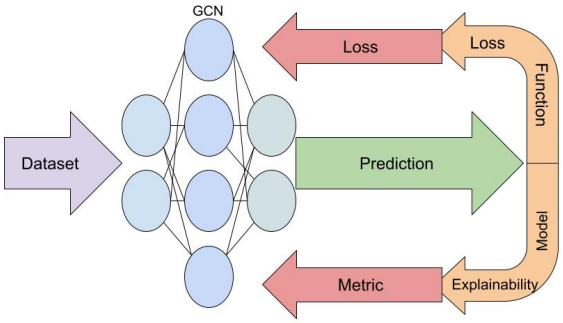

FIGURE 3. Pipeline for training the model with Additional Loss Functions.

For each epoch, the proposed framework involves computing the Categorical Cross-Entropy Loss (CCL) of the GNN, then the explanation vectors from a given XAI for all nodes in the graph. Afterwards, the explanations for each node are evaluated in regard to a given metric. This is then added to the CCL and backwards propagated through the Neural Network. For instance, to optimise the stability of explanations from Integrated Gradients we would, for each epoch, compute the CCL then compute Integrated Gradients explanations for all nodes and finally the stability of these explanations. We then add the stability to the loss and backwards propagate. This is reflected in Figure 3, and Algorithm 1 of the appendix.

Here, we train a 2 layer graph convolutional network with 16 and n hidden nodes respectively for 100 epochs (n being the number of classes for each dataset). Between each layer, is a ReLU activation function and dropout function. After the final layer there is a softmax activation function. We use Adam as the optimizer with a learning rate of 1e-02 and weight decay of 5e-04. For the implementation of Integrated Gradients and Class Activation Maps, we use the associated functions from [1]. The implementation of the XAI quality metrics was based on the functions provided in section 2.2. To ensure repeatability, we seed the Neural Network with 12345 in Py-Torch.

4 Results

To evaluate the research questions defined in Section 1, we first investigate the optimisation of sparsity for Integrated Gradients and Class Activation Maps explanations, then stability and finally sufficiency.

4.1 Sparsity

TABLE 1. Original (O) and Final (F) accuracy and sparsity of GNN's trained to optimise sparsity.

	XAI	O. F1	F. F1	O. Sparsity	F. Sparsity
Cora	IG	0.804	**0.944**	1.000	1.000
	CAM	0.804	0.811	1.000	**0.802**
Citeseer	IG	0.680	**0.878**	0.997	0.997
	CAM	0.680	0.684	1.000	**0.917**

In Table 1 we find that GNN's trained to optimise IG sparsity display not only no reduction in accuracy, but a 14.4% and 19.8% macro-F1 increase on Cora and Citeseer respectively. Yet, the sparsity of IG explanations for the trained models was unable to be optimised on both Cora and Citeseer.

For CAM, however, both an accuracy increase and sparsity reduction is present, indicating that although IG explanations cannot be made more sparse, CAM explanations can be more sparse. As such, we find that the locality of a CAM explanation can be improved through Additional Loss Functions.

Moreover, the best macro-F1 achieved when training to optimise the sparsity of explanations was 94.4% by the GNN trained for IG sparsity, on Cora. The aforementioned accuracy is c. 4% higher than the latest benchmark on Cora by [8], and 14.4% greater than the initial accuracy of the GNN.

4.2 Stability

TABLE 2. Original (O) and Final (F) accuracy and stability of GNN's trained to optimise stability.

	XAI	O. F1	F. F1	O. Stability	F. Stability
Cora	IG	0.804	**0.835**	13.22	**6.208**
	CAM	0.804	0.811	387.8	162.5
Citeseer	IG	0.680	0.680	12.12	**10.69**
	CAM	0.680	**0.684**	154.1	36.19

For the GNN's trained to optimise the stability of explanations from Integrated Gradients and Class Activation Maps we

find a larger reduction in stability than sparsity experienced. For the IG models we observe a reduction in the distance between explanation matrices of a perturbed and non-perturbed graph, which indicates that the GNN learnt to disregard randomly added nodes when forming its decision. This is consistent in the CAM models, although, the reduction is more significant than for IG models.

Cora and Citeseer, however, are sparse graphs where the average subgraph of a node is small. As such, the random addition of two nodes, as was the case when calculating stability, significantly alters the subgraph. Furthermore, it is apparent that two nodes are randomly added if the subgraph is small, and this may be why the GNN improved the stability of explanations through Additional Loss Functions.

4.3 Sufficiency

TABLE 3. Original (O) and Final (F) accuracy and sufficiency of GNN's trained to optimise sufficiency.

	XAI	O. F1	F. F1	O. Sufficiency	F. Sufficiency
Cora	IG	0.804	**0.941**	0.352	**0.067**
	CAM	0.804	0.811	0.595	0.582
Citeseer	IG	0.680	**0.878**	0.413	**0.282**
	CAM	0.680	0.681	0.608	0.569

As per Table 3, IG reported a substantial accuracy increase and sufficiency reduction for both Cora and Citeseer. This indicates that a GNN can be trained to place importance on a smaller number of nodes in another node's classification. Like IG, CAM also reported a reduction in sufficiency but only a minimal accuracy improvement. It is proposed that IG is more representative of the GNN than CAM due to its lower initial sufficiency and ability to more substantially improve the sufficiency of its explanations.

Furthermore, IG sufficiency reported a similar accuracy increase to the GNN trained to optimise IG sparsity. This again 4% above the latest benchmark, and reaffirms the potential use of training for both IG sufficiency and sparsity.

5 Discussion

In Section 4 we determined that both accuracy and explanation quality can be improved through Additional Loss Functions. Here, we investigate the modus operandi behind the quality and accuracy improvements.

5.1 Explanations

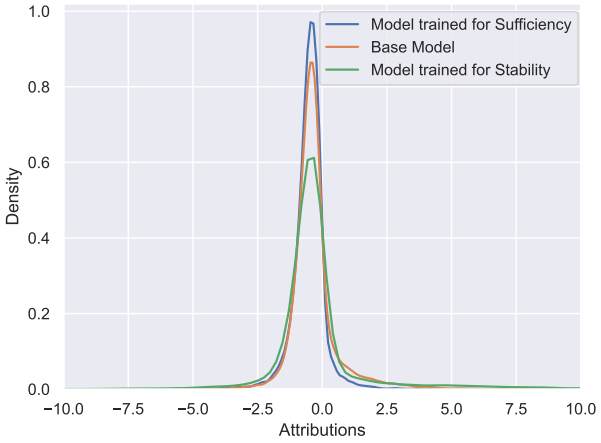

FIGURE 4. Integrated Gradients attributions from GNN's trained to optimise IG quality, on CORA.

To investigate the effect of Additional Loss Function training, we generate all explanations for all nodes on a given dataset by Integrated Gradients, and then plot these explanation values. We see this in Figure 4, where the explanations from all GNN's trained to optimise IG stability and sufficiency are plotted. We find that the distributions between IG explanations are statistically significantly different with a Kolmogorov-Smirnov p-value < 0.05. We also find that mean IG attributions for the base model, the model trained to optimise stability, and the model trained to optimise sufficiency were -0.31, -0.51 and -0.11 respectively. This shows that, when training to increase the faithfulness of explanations, the mean explanation will converge towards zero. This is intuitive as if you keep only the top q% nodes in another node's classification, all other nodes should have an attribution of zero. Further, the model trained to optimise stability has the greatest standard deviation at 2.2, suggesting that the GNN learnt to disregard the randomly added nodes, and increase the importance placed on other nodes.

As the distributions of IG attributions are significantly different for the base GNN, GNN trained to optimise IG stability, and GNN trained to optimise IG sufficiency, we investigate the distribution of CAM attributions for those same models. This is aimed at testing whether a GNN trained for one XAI, can provide different attributions for another XAI.

We observe that the base GNN's CAM explanations tend towards zero more than the GNN's trained for IG quality. We see that the GNN trained for IG sufficiency has the largest CAM

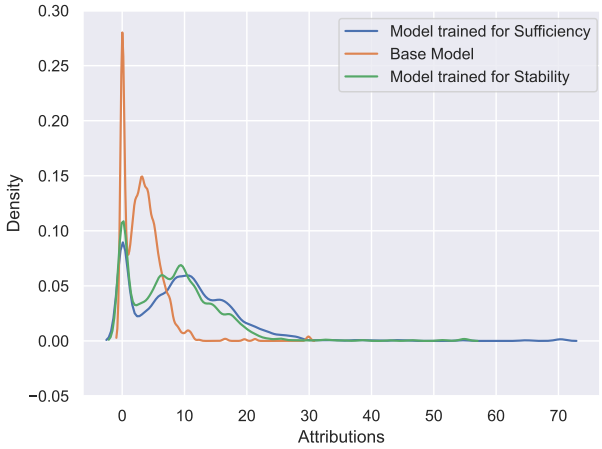

FIGURE 5. CAM attributions for GNN's trained to optimise IG explanation quality, on CORA.

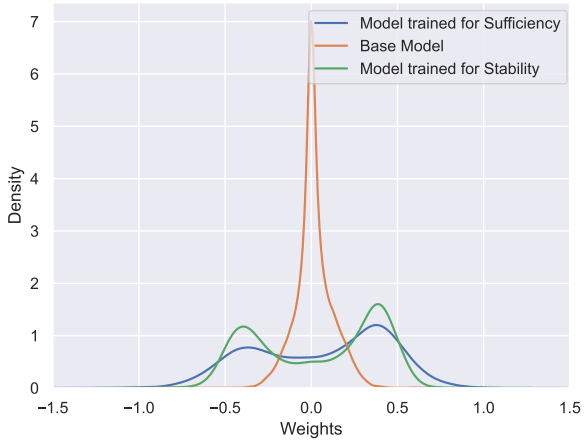

FIGURE 6. Model weight distribution of GNN's trained to optimise IG explanation quality, on CORA.

explanation range, indicating that the GNN learnt to prioritise the importance of nodes it considers truly important. This differs from the reduction in magnitude and variance of IG explanations from the GNN trained to optimise IG sufficiency, in Figure 4. For the GNN trained to optimise IG stability, however, we observe a greater variation than the base GNN, but whose values tend more towards zero than the GNN trained for IG sufficiency. This again contrasts Figure 4, and suggests that the effects of GNN's trained to optimise IG and CAM explanation quality may be antagonistic, and thus make it impossible to train for the quality of explanations from both IG and CAM.

5.2 Model Parameters

To understand the effect of Additional Loss Function training on the underlying GNN's structure, we extract the model weights for the baseline GNN and the GNN's trained to optimise IG stability and sufficiency. We plot the weight distribution in Figure 6. We find that GNN's trained to optimise explanation quality prefer a bimodal model parameter distribution, instead of the mono-modal distribution of the base model. This shows that the underlying model is materially affected by Additional Loss Functions. As the GNN's with bimodal model parameter distributions perform significantly better in both macro-F1 and explanation quality than the base model, this suggests that a bimodal model parameter distribution may induce not only a better accuracy but a GNN that will provide higher quality explanations.

To understand why the GNN's trained for IG sparsity and sufficiency attained such drastic macro-F1 increases the network was deep copied during the calculation of Integrated Gradient's sparsity. This ensured that there was no path through the network, such that the backwards propagated values would be propagated twice through the GNN. When this occurred, the GNN's accuracy was not increased, and sparsity was unchanged. This indicated that the accuracy improvement from Integrated Gradients was a function of the loss being backwards propagated multiple times. Absurdly, when backwards propagating the initial loss multiple times, without integrated gradients, there was an insignificant change to the Neural Network's accuracy. This indicates that the backwards propagation of loss through Integrated Gradients provides an accuracy improvement to the model.

6 Conclusion and Future Work

Through injecting Additional Loss Functions into the training of a Graph Neural Network, on Cora and Citeseer, we have found that model accuracy can be significantly improved and the quality of explanations on the model can be concurrently improved. We have empirically proved that, sparsity cannot be optimised for Integrated Gradients, whereas stability and sufficiency can be improved through additional loss factor training. For Class Activation Maps, we have determined that all metrics can be improved, but with a lower accuracy improvement than Integrated Gradients provided. In the future, we hope to extend these findings to more combinations of XAI and XAI metrics in the graph space. We also seek to affirm the use of Addi-

tional Loss Functions in computer vision and natural language processing. The usefulness of modern XAI resides on the provision of trustworthy explanations. If an operator cannot trust the XAI, they cannot trust the underlying Neural Network. As such, the improvement of XAI quality directly influences the usefulness of XAI in Machine Learning.

7 Appendix

Algorithm 1 Additional Loss Function Algorithm

Require: E ▷ Explainability model,
F ▷ Neural Network,
G ▷ Metric Function,
x ▷ Input,
y ▷ Labels,
L ▷ Loss Function,
EPOCHS

Ensure: $EPOCHS \geq 1$
while $\epsilon \in EPOCHS$ **do**
$pred \leftarrow F(x)$
$loss \leftarrow L(pred, y)$
$e \leftarrow E(F, x, y)$
$m \leftarrow G(e, F, x, y)$
$loss \leftarrow loss + m$
$F.backwards\_propagate(loss)$
end while

References

[1] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks, 2023.

[2] Chirag Agarwal, Marinka Zitnik, and Himabindu Lakkaraju. Probing gnn explainers: A rigorous theoretical and empirical analysis of gnn explanation methods, 2022.

[3] Btd. Explainable ai's (xai) explainability metrics: Quantifying the interpretability of models, Jan 2024.

[4] Chun Sik Chan, Huanqi Kong, and Guanqing Liang. A comparative study of faithfulness metrics for model interpretability methods, 2022.

[5] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics.

[6] Nathan O. Hodas and Panos Stinis. Doing the impossible: Why neural networks can be trained at all. *Frontiers in Psychology*, 9, 2018.

[7] IBM. What is explainable ai (xai)?, 2024.

[8] Mohammad Rasool Izadi, Yihao Fang, Robert Stevenson, and Lizhen Lin. Optimization of graph neural networks with natural gradient descent, 2020.

[9] Vivek Miglani, Narine Kokhlikyan, Bilal Alsallakh, Miguel Martin, and Orion Reblitz-Richardson. Investigating saturation effects in integrated gradients, 2020.

[10] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2020.

[11] Legal Office of European Union. About us: Explainability notice, 2024.

[12] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[13] Cynthia Rudin and Joanna Radin. Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition. *Harvard Data Science Review*, 1(2), nov 22 2019. https://hdsr.mitpress.mit.edu/pub/f9kuryi8.

[14] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019.

[15] Timo Speith. How to evaluate explainability? - a case for three criteria. In *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*, pages 92–97, 2022.

[16] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[17] Giulia Vilone and Luca Longo. Explainable artificial intelligence: a systematic review, 2020.

[18] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. Adversarial examples on graph data: Deep insights into attack and defense, 2019.

[19] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey, 2022.

[20] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015.

[21] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.